

# Saliency Driven Object recognition in egocentric videos with deep CNN

Electronic copy submitted to the Journal of Computer Vision and Image Understanding

Philippe Pérez de San Roman<sup>a,b</sup>, Jenny Benois-Pineau<sup>a,b,\*</sup>, Jean-Philippe Domenger<sup>a,b</sup>, Florent Paclet<sup>a,c</sup>, Daniel Cataert<sup>a,c</sup>, Aymar de Rugy<sup>a,c</sup>,

<sup>a</sup> *Université de Bordeaux, 35 Place Pey Berland, 33000 Bordeaux, France*

<sup>b</sup> *LaBRI - UMR 5800 - CNRS Université de Bordeaux, IPB, 351, cours de la Libération, 33405 Talence, France*

<sup>c</sup> *INRIA - UMR 5287 - CNRS Université de Bordeaux, 146, rue Lo Saignat 33076 Bordeaux cedex, France*

---

## Abstract

The problem of object recognition in natural scenes has been recently successfully addressed with Deep Convolutional Neuronal Networks giving a significant break-through in recognition scores. The computational efficiency of Deep CNNs as a function of their depth, allows for their use in real-time applications. One of the key issues here is to reduce the number of windows selected from images to be submitted to a Deep CNN. This is usually solved by preliminary segmentation and selection of specific windows, having outstanding "objectiveness" or other value of indicators of possible location of objects. In this paper we propose a Deep CNN approach and the general framework for recognition of objects in a real-time scenario and in an egocentric perspective. Here the window of interest is built on the basis of visual attention map computed over gaze fixations measured by a glass-worn eye-tracker. The application of this set-up is an interactive user-friendly environment for upper-limb amputees. Vision has to help the subject to control his worn neuro-prosthesis in case of a small amount of remaining muscles when the EMG control becomes unefficient. The recognition results on a specifically recorded corpus of 151 videos with simple geometrical objects show the mAP of 64,6% and the computational time at the generalization lower than a time of a visual fixation on the object-of-interest.

**Keywords:** Psycho-visual attention, Saliency, Egocentric video, Object recognition, Deep Convolutional Neural Networks

---



---

\*Corresponding author

*Email addresses:* philippe.perezdesanroman@labri.fr (Philippe Pérez de San Roman), benois-p@labri.fr (Jenny Benois-Pineau), domenger@labri.fr (Jean-Philippe Domenger), florent.paclet@u-bordeaux.fr (Florent Paclet), daniel.cattaert@u-bordeaux.fr (Daniel Cataert), aymar.derugy@u-bordeaux.fr (Aymar de Rugy)

## 1. Introduction and motivation

The problem of natural object recognition in images has been in the center of computer vision community since quite a lot of time. Previous PascalVOC challenge [1] and ongoing ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [2] have united important task forces for finding the solution of this problem in natural visual scenes. Since recently, developed approaches find their place in a quite realistic application for health care and patient’s monitoring such as in [3]. The pioneering works on real-world manipulated object recognition in egocentric perspective [4] for evaluation of cognitive impairment of Alzheimer patients [5] showed that even if the state-of-the art in egocentric object recognition does not allow for getting scores approaching 100% for all object categories and requires a heavy annotation process, the information on the object of interest the human interacts with is essential for assistance and evaluation of patients and impaired subjects. In this paper we develop an object recognition approach for assistance of upper-limb amputees wearing neuro-prostheses.

Classic myoelectric control of neuro-prostheses uses the activity of the remaining muscles to control the multiple degrees of freedom of the prosthesis. This strategy, however, faces a fundamental problem related to fact that the higher the amputation, the higher the number of degrees of freedom of the prosthesis to control with less control signals from the fewer remaining muscles. In addition, all commercially available myoelectric prosthesis are only controlled with two muscle groups, one flexor and one extensor, and therefore involve sequential control of individual joint with unnatural control schemes to switch between joints. The tedious learning for relatively mediocre results associated with these control schemes has motivated several laboratories at developing control schemes that integrate a higher number of muscles, either through pattern recognition of movement classes from muscle recordings [6, 7], or through regressions that use muscle activities for simultaneous proportional control of the multiple degrees of freedom of the prosthesis [8, 9, 10]. Despite their merits, however, these attempts do not resolve the key limitation in terms of number of remaining muscles, although some control signals could be recovered using highly invasive surgical techniques such as nerve recording [11] or targeted muscle re innervation [12].

Promising and much less invasive alternatives propose to use other control signals such as from computer vision [13, 14], gaze information [15, 16], and/or residual biological motion [17]. In [13], stereo-vision from camera integrated in augmented reality glasses was used to automatically select grasp type and size from a visible object, and [14] added inertial sensing to automatically align wrist orientation to that of the object to grasp. In both instances, however, only one object was present in front of the subject, thereby avoiding the critical problem of recognizing the object of interest from the multiple objects typically present in a natural environment. Furthermore, although camera on glasses provided egocentric videos, gaze information was not available and therefore not used to assist this process. In [15, 16], gaze information was used, but to supplement muscle recordings for the control of reaching actions rather than to recognize

the object of interest. Furthermore, the setup was such that eye tracking was working on a fixed head, and the reaching was on a two dimensional screen. Our goal here is to combine gaze information with recent progresses in deep learning for computer vision, in order to improve real time object recognition from egocentric videos, the long term goal being to incorporate this information into prosthetics control. The rest of the paper is organized as follows. In Section 2 we analyse the related works in natural object recognition and localization with Deep Convolutional Neural Networks and summarize our contributions, in Section 3 we present our general approach for saliency driven object recognition with Deep CNN, which uses gaze information. In Section 4 we focus on Network design and tuning. Section 5 presents experiments and results. In Section 5 discussion, conclusion and further perspectives of this work are presented.

## 2. Related Works

The problem we adress consists in both: i) object recognition and ii) object localization. In [18] a good analysis of recent approaches for object localization has been proposed, such as "regression approaches" as in [19][20], and "sliding window approaches" as in [21] when the CNN processes multiple overlapping windows. The authors of [18] propose a so called Region-based convolutional network (R-CNN). Inspired by "selective search" approach [22] it evaluates multiple (2K) "object proposals" and finally train an SVM for classification. Such "proposals" are numerous and the multi-task classification with CNNs requires heavy computations. Hence in [18] the authors report that R-CNN can take from 10s to 45s for object classification. Several attempts has been made to accelerate the generalization process. In [23] a weakly supervised training scheme is designed. They train the network on the whole labeled image. The deepest layers of the CNN supply features. Then fully connected layers - adaptation layers are proposed considered as convolution layers. Max pooling supplies the scores for different positions of objects. In order to reduce the number of object proposals, the spatial grouping of windows was proposed in SPPnet[24]. Here the spatial pyramid pooling layer is added above the convolutional layers, transforming the output into fixed-size vectors. Thus built the network not only copes with different sizes of input images, but is from 24 to 54 times faster than AlexNet [25] with only 530 ms per image. Further acceleration at testing step is proposed in fast R-CNN [26]. Here the idea of training on the whole image is re-used. The network first processes the whole image at the deep layers of the network and then, at the uppaer layers, object proposals are processed. Indeed the whole image is used through several convolution and max pooling layers instead of object proposals. The latter are used at the so-called region-of-interest (ROI) pooling layer. Here a fixed-lentgh feature vector is extracted for each object proposal from the obtained full-image feature map. The feature vectors are then submitted into a sequence of fully connected layers, finally two output layers produce the softmax probability for object classes and the background and the estimate of window corner positions with regression.

The speed-up of computation at training step is achieved due to the late selection of features for object proposals, while at the test step, they use truncated SVD decomposition on fully connected layers to accelerate computations for them. The computational time of 600ms is reported per image at the test step. All these methods developed for object recognition and localization with Deep CNN aim at increasing mAP and reducing computational time. They work in an "unconstrained" setting, which means that there is no initial assumption on object location. Thus the localization process has to be accelerated. One of the trends strongly present in the research in object recognition which also inspired "selective search" [22] approach, consists in using visual saliency of regions, which serves to select object proposal candidates. Out of the methodological framework of Deep CNN, such methods were proposed for the popular Bag-of-Visual-Words (BoVW) model[27]. The latter served as object signature but was built on the whole image from quantized features weighted by underlying saliency value[28], [29]. Despite we have developed a complete saliency-based methodology for all steps in feature engineering approach, such as feature selection, encoding, pooling, and despite its good performances surpassing the most popular state-of-the-art model DPM [30], the BoVW approach showed its limits even with ideal saliency maps such as manually annotated bounding boxes [31]. As the predicted (objective) or ideal (subjective), computed with gaze-fixations from eye-tracker recording, maps confine image analysis process and eliminate clutter, it is natural to try to incorporate them into the winner model today, such as Deep CNN.

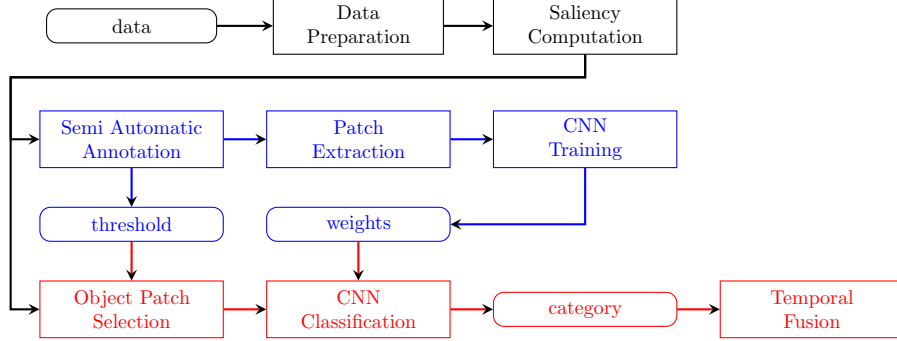
The contributions of our work are the following: i) we introduce ideal saliency maps, recorded with head-mounted eye-tracker into "object proposal" selection for our object recognition need in an interactive environment. Taking into account that object recognition has to be conducted, on acting person, we use established facts from cognitive sciences and bio-physics to select sequences from video and filter-out distractors ii) we re-use "ImageNet" architecture, propose adequate data sampling and augmentation relevant to our egocentric setup and show that without any supplementary efforts, the base-line ImageNet allows to get rather good scores for selected object proposals and do it in a biological real-time - shorter than visual fixation time.

In the following section we present our general framework for saliency driven object recognition with Deep CNN.

### **3. General approach for saliency driven object recognition with Deep CNN**

Our method is developed for humans grasping an object. The task consists in simultaneous recognition of the object the subject wishes to grasp. A human subject is instrumented with a glass-worn eye-tracker with a scene camera (Tobii Pro glasses 2). The block diagram of the method is presented in figure 1. The data preparation block serves to filter out missing eye-tracker recordings, see Section 3.2. On the basis of recorded gaze fixations we compute visual saliency of pixels in the video recorded by the scene camera, see Section 3.3.

Figure 1: Block diagram of our method, the upper-branch blocks are common for training and test, the middle branch is the training scheme, and the lowest branch is the online processing.



In real world scenario our system fulfills automatic patch selection, i.e. "object proposal" (see the lowest branch of figure 1) using computed saliency. Then CNN classifies the extracted patch into a set of known categories. Fusion of classification scores along the time allows for filtering natural noise such as eye-blinking and saccadic motion towards distractors. The middle branch of the diagram in figure 1 presents training process. To train the known category of objects we use a semi-automatic annotation method, also guided by computed saliency, see Section 5.2. Specific data augmentation approach is designed with regards to the real life scenario, see section 4.2.2. CNN training is realized on the augmented object dataset. In the following part of this section we detail these steps.

### 3.1. Physiology of visual attention

In order to provide rationale for our data preparation methodology we briefly expose here our physiological hypotheses and the known neuro-physiological models of human vision relevant to our problem. Our actor, an upper-limb amputee has less freedom in the control of his body than a healthy person. When he wants to grasp an object in his environment he first looks at it (which is not always the case for a healthy subject). This is our main assumption.

The observation of a scene comprises: (1) The *discovery* of the scene where the eye scouts sparsely the scene. (2) The *fixation* on the object of interest. (3) *Micro-saccades*, when the eye slightly oscillates about the target object. (4) *grasping movement* is triggered. (5) Also some *distractors*, such as audio; light; motion; and occlusions in the scene can lead the eye to deviate to another object.

We have conducted psycho-visual experiments on healthy volunteers aged from 20 to 23 and we observed that: (1) The *scene discovery* takes from 240 to 300 ms. (2) The *fixation* is about 250 ms. (3) *Micro-saccades* can occur with duration about 6 to 300 ms according to different sources reviewed in [32], note that the frequency of our eye-tracker does not allow precise measurements of micro-saccades duration, see section 5.1.1 for experimental set-up. (4) The

*grasping movement* then takes between 400 to 900ms. **(5)** Finally the times of *distractor fixations* are between 100 and 500 ms. These data are in accordance with the results in [33], [34].

During the *scene discovery* a subject explores the scene searching for the target object, hence the object-of-interest is not fixated. Therefore we reject the beginning of each video sequence when selecting both training and validation frames and do this at the test stage as well. *Micro-saccades* are not a problem thanks to the interpolation of fixation coordinates that maintain the eye fixation on the object-of-interest. *Distractors*, nevertheless, are a real challenge: they cannot be automatically identified neither when performing semi-automatic annotation for training nor at the test stage. And so they are included in our training set in the form of incorrectly labeled patches. To deal with them in our online framework, we propose to use temporal fusion of the classification results that filters out frames with distractors.

### 3.2. Data Preparation

The data preparation block receives the data from the Tobii Glasses 2 streams, simulated in the present work as two pre-recorded files: one for the video and one for the eye-tracking data. In the eye-tracking data some recordings of gaze fixations are missing due to eye-blinking of the subject. Also the video and the gaze tracking have different sampling rates, and so it is rare that an eye-tracking record is synchronized with a video frame.

To cope with all these problems, we apply interpolation (using a spline). This smoothes the gaze fixation data and allows us to synchronize the two streams.

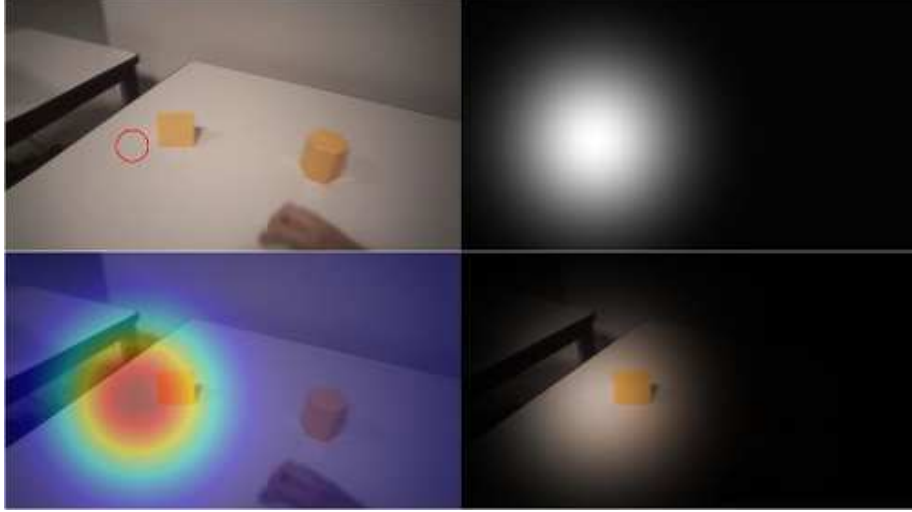
### 3.3. Subjective saliency Computation

From the eye-tacker data we get the recorded gaze-fixation point  $f$  for the image  $I$ , with coordinates  $x_f$ ,  $y_f$  and  $d_f$ . Here  $d_f$  is the coordinate along the axis of gaze direction and  $x$ ,  $y$  are the coordinates of the fixation point in the image plane of video recorded with the scene camera of glasses. Therefore, the so-called "subjective saliency map" or Wooding's map  $W(I, f)$  [35] can be computed. It is a normalized Gaussian function, centered on a fixation point, with values close to 1 in the vicinity of the fixation point (in the focal vision), and values close to 0 in pixels situated far from it (in the peripheral vision). The spread  $\sigma(d)$  of the Gaussian is adapted to the size of the image  $I$  and the distance  $d_f$  to the object to model the focal vision. It is also normalized to sum to 1. A visual example is given in figure 2. The equations 2 below detail the computation of the Wooding Map, and its parameters:

$$\sigma(I, d) = \frac{A}{d} \cdot \frac{\text{width}(I)\tan(180\alpha\pi)}{2\tan(\beta\pi/180)} \quad (1)$$

$$W(I, f, x, y) = \frac{A}{\|W\|_\infty} \times \exp \frac{-(x - x_f)^2 - (y - y_f)^2}{2 \cdot \sigma(I, d_f)^2 + \epsilon} \quad (2)$$

Figure 2: Different forms of Wooding’s map (in raster scan order) (i) The original image  $I$  with the fixation point  $f$  drawn in red on it and (ii) Normalized Wooding’s Map  $W(I, f)$  (iii) The Heat-Map visualization; (iv) The Weighted Map where the saliency is used to weight brightness in the frame.



Where  $\alpha = 2^\circ$  is the angle of projection of the fovea,  $\beta = 24^\circ$  is the camera opening angle on the width,  $A = 1600$  mm is the maximum distance to an object according to our setting and  $\epsilon$  is a small number  $\epsilon = 0.01$ .

In our case of moving eye-tracker wearer, unlike the traditional Wooding’s map, the spread  $\sigma$  of the Gaussian function is linearly adapted wrt the distance to the object. Thus our method produces a larger image patch when the object is closer (and so appear larger in the video), or reciprocally, smaller image patch when the object is farther. In figure 2 we show different forms of Wooding’s saliency map on a video frame.

#### 4. Network design and tuning

In our work we used the basic ImageNet architecture proposed in [25]. We do not need a specific optimization of computational cost as the focused selection of an “object proposal” accordingly to the terminology of Girshik [26], in each frame allows us to be compatible with real-time requirements for object recognition. In this section we remind the architecture and focus on the way we extract “object proposal” patches and the background candidates based on saliency, and then augment them to prevent over-fitting.

##### 4.1. General architecture and parameters

*Network layers.* The ImageNet network [25] is mainly composed of 5 types of layers: 5 convolutions (Conv), 3 fully connected (FC), 7 Rectified Linear

Table 1: ImageNet [25] architecture.  $k$  is the kernel size,  $nb$  is the number of filters learned,  $b$  is the bias,  $pad$  is the zero-padding size, and  $s$  the stride

Layer	Depth	Type	Name	Parameters	Top shape
23	8	Soft Max	prob		C
22	8	FC	ip8		C
21	7	Dropout	drop7	$ratio = 0,5$	4096
20	7	ReLU	relu7		4096
19	7	FC	ip7	$b = 1$	4096
18	6	Dropout	drop6	$ratio = 0,5$	4096
17	6	ReLU	relu6		4096
16	6	FC	ip6	$b = 1$	4096
15	5	Max pooling	pool5	$k = 3 \times 3, s = 2$	6x6x256
14	5	ReLU	relu5		13x13x256
13	5	Convolution	conv5	$k = 3 \times 3, nb = 256, pad = 1, b = 1$	13x13x256
12	4	ReLU	relu4		13x13x384
11	4	Convolution	conv4	$k = 3 \times 3, nb = 384, pad = 1, b = 1$	13x13x384
10	3	ReLU	relu3		13x13x384
9	3	Convolution	conv3	$k = 3 \times 3, nb = 384, pad = 1, b = 0$	13x13x384
8	2	LRN	norm2	$k = 5 \times 5, \alpha = 10^{-4}, \beta = 0,75$	13x13x256
7	2	Max pooling	pool2	$k = 3 \times 3, s = 2$	13x13x256
6	2	ReLU	relu2		27x27x256
5	2	Convolution	conv2	$k = 5 \times 5, nb = 256, pad = 2, b = 1$	27x27x256
4	1	LRN	norm1	$k = 5 \times 5, \alpha = 10^{-4}, \beta = 0,75$	27x27x96
3	1	Max pooling	pool1	$k = 3 \times 3, s = 2$	27x27x96
2	1	ReLU	relu1		55x55x96
1	1	Convolution	conv1	$k = 11 \times 11, nb = 96, s = 4, b = 0$	55x55x96
0	0	Data	data		227x227x3

Units (ReLU), 3 Max pooling, and 2 Local Response Normalization. They are combined vertically to increase the network depth.

1. The convolution is used to extract features on its input by applying filter to it. On the first layer this filters respond to edges or color blobs, while on the last one they are able to abstract shapes and objects parts [25].
2. The FC layers are used to progressively map activation maps to a single dimension feature vector where at the end each value is associated to a class of object. It is then normalized to a probability distribution using a Soft Max layer.
3. The max polling [25] is used to spatially down-sample the activation of the previous layer by only propagating the maximum activation of a previous group of locally connected neurons.
4. Local Response Normalization is used to normalize the response of neurons at the same spatial location. This is inspired by lateral inhibition in real neuron [25].
5. ReLU were introduced in [25] to increase the network optimization convergence.

*Optimization.* We use the soft max loss function (multinomial logistic loss) already implemented in Caffe [36], that for input image  $X_i$  with known label  $l_i$  is:

$$E(X_i, l_i) = -\frac{1}{N} \sum_{j=1}^N \log(\hat{P}_j) \delta(\hat{l}_j, l_i) \quad (3)$$

$$\delta(l, l_i) = 1 \text{ if } l = l_i, 0 \text{ else} \quad (4)$$



Where  $\hat{P}_j$  is the probability with associated label  $\hat{l}_j$ , resulting from the forward pass in the network. The overall loss over a dataset  $D$  is:

$$L(W) = \frac{1}{|D|} \sum_{i=1}^{|D|} E(X_i, l_i) + \lambda r(W) \quad (5)$$

$r(W)$  is a regularization term with weight decay  $\lambda = 0.0005$

We use the default implementation of stochastic gradient descent from Caffe and ImageNet [25, 36] with weight update rule:

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t) \quad (6)$$

$$W_{t+1} = W_t + V_{t+1} \quad (7)$$

Where  $\alpha$  is the Learning rate and  $\mu$  the Momentum (0.9). Our learning rate  $\alpha$  is initialized to 0.001 and decreased by half every 30000 iterations.  $\nabla L(W_t)$  is computed through back-propagation in the network.

## 4.2. Saliency-based data preparation

We will now present the preparation of the input data for the network training, which consists in selection of bounding boxes of objects using saliency maps, and sampling of background patches. We also propose a data augmentation strategy corresponding to our problem.

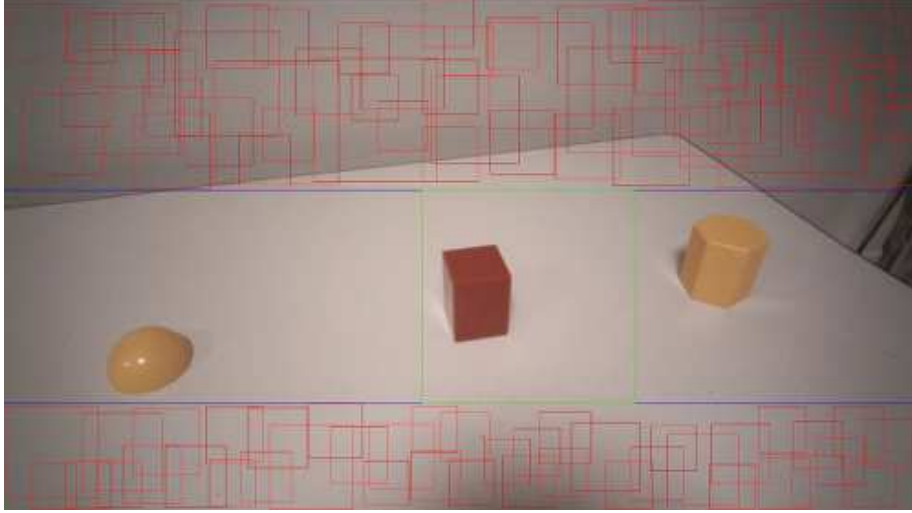
### 4.2.1. Patch extraction

Our basic CNN [25] has a fixed input resolution, so no matter at which resolution the objects appear in the video, they are all resized to a fixed size (227x227 RGB in our case). For machine learning, especially classification, we have to extract object patch (examples) for all categories, including the background that is the rejection category. It is important to extract a similar amount of examples for each of them to avoid imbalanced class problem.

In our scenario, the object of attention is identified by the thresholded saliency, and a label describing the category. We extract the corresponding blob by connected component analysis, it gives us the bounding box. The green bounding box in figure 3 is an example of an image patch corresponding to a "rectangular prism". We also have to extract background at the same time to ensure that we have the same amount of rejection examples. Remember that our experimental setup specifies that the objects are lined-up on the table. Since only one object is labeled, we exclude the bounding box of the object, but also the area where other objects could appear, to avoid a background/object mixture. This exclusion area is drawn in blue in the figure 3 below.

Background patches are then sampled randomly in the remaining parts of the image. Their minimum resolution is limited to 95x95 pixels induced by full HD resolution of our videos (1080x1920). When sampling several background patches in the same image, we respect the maximal overlap of 20%. Thus we ensure sampling of different areas in the image background and therefore, we capture more information on it. The figure 3 shows many random background patch proposals. In practice we keep only one or two per video frame in order to avoid an imbalanced class problem. Due to the random sampling and the repeatability of the background in video, we extract samples well covering the background of video scenes.

Figure 3: Patch extraction: (i) Middle: bounding box of the object-of-interest (ii) Middle left and right: the exclusion area. (iii) Top and bottom: example of background patches with a maximum overlap of 20%



#### 4.2.2. Patch Augmentation

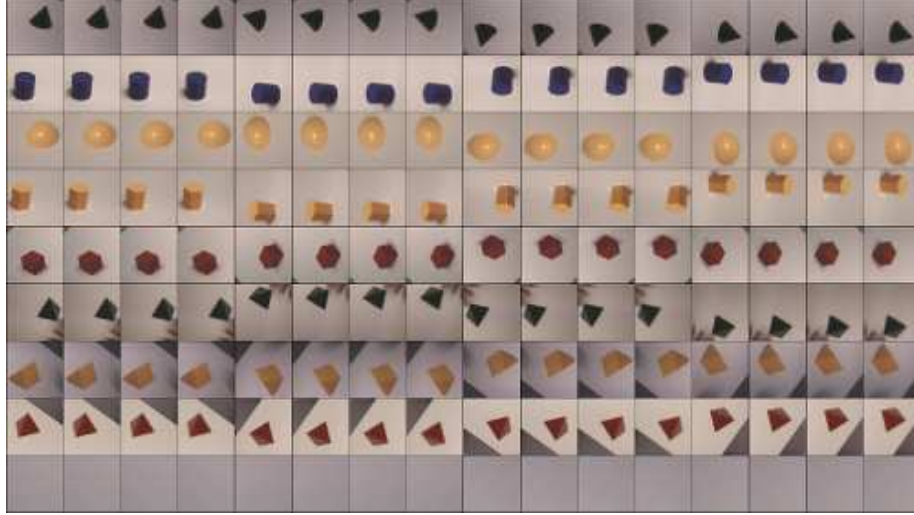
Data augmentation is very efficient to prevent over-fitting [25]. The idea is to apply label-preserving transformation to the image patch, and give both the original one and the transformed ones to the network for training. This will artificially increase the training dataset size. Common transformations are horizontal mirroring, random cropping [36]. In our case objects can be placed upside-down or lay on any of their sides, this led us to rotate training image patch by an angle  $\alpha \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ . We only considered multiples of  $90^\circ$  to avoid discretization problems that can lead to a drastic accuracy drop, due to the parasite high-frequency components in the image spectrum. As the video can be blurred by fast motion of the glass-worn camera, which is often the case in egocentric videos, we decided to blur training image patches by 3 Gaussian kernels of size  $k \in \{1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7\}$ . This increases the network robustness to motion blur. In total we increase our training by 16  $((4 \times N) \times 4)$  (the rotation by  $\alpha = 0^\circ$  followed by a blurring  $k = 1 \times 1$  leave a patch unchanged). We also apply this transformation to background image patches to preserve class balance. The figure 4 shows an example for all object categories.

Note, that we do not need data augmentation at the test step as it is proposed in ImageNet [25]. They need it as they do not have certainty on the object location. This is why they generate multiple candidates for the "object proposal" and practice fusion of scores. In our case an "object proposal" is unique as it is totally defined by online recorded gaze fixation and derived saliency maps.

#### 4.3. Temporal Fusion

We are classifying a sequence of object proposals in a video with "mean fusion" operator, which is equivalent to a simple sum as shown in the equation below 8: The scores for each class are summed over the candidate patches along a video, and the

Figure 4: Patch augmentation: Rows show different object categories and column show different augmentations. Each group of 4 columns depict different rotation angle  $\alpha$ , and within this, each column is a different blurring kernel size  $k$



final category is the one that obtains the maximum score over these sums.

$$c(V) = \underset{c}{\operatorname{argmax}} s(V, c) = \underset{c}{\operatorname{argmax}} \left\{ \sum_{i=1}^N s(p_i, c) \right\} \quad (8)$$

Where  $V$  is a video,  $p_i$  is the candidate patch of frame  $i$ .

We use this instead of retaining just the most frequent classification result, i.e. the "majority vote", because we believe the score of an incorrectly classified patch is often much smaller than the score of a correctly classified patch. And so by summing the scores over a patch, the correct class score wins.

## 5. Experiments and results

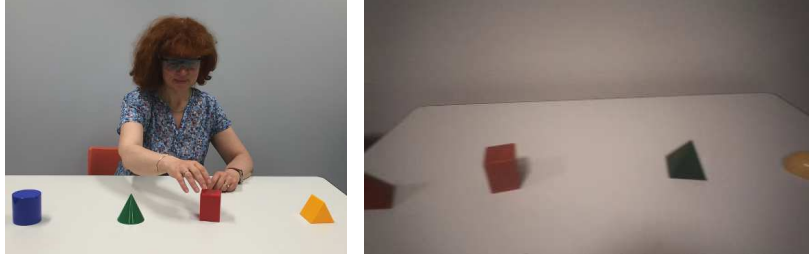
In order to test our object recognition framework in a real life but simplified scenario we produced a new dataset that we called Large Egocentric Gaze Objects (LEGO) which will be soon available online. Below we present the experimental setup and content; our data selection results, network optimization, and results.

### 5.1. LEGO Dataset

#### 5.1.1. Experimental Setup

The recording of our dataset was conducted with four healthy volunteers aged from 20 to 23. In each recording session a subject was instructed to look for a specific object and to grasp it. The subjects were sitting in front of a white table, facing a white wall. They wore the Tobii Pro Glasses 2. This eye-tracker records gaze fixations at 50 Hz and video frames at 25 Hz. At first, the subject's eyes were closed (in which case the gaze data are not available). Four objects out of eight different objects were randomly

Figure 5: Left: A subject equipped with the Tobii Pro Glasses 2 performing the experiment. Right: The egocentric field of view of the glasses.



chosen and placed in line on the table. They were presented in different positions for each experiment (they could be placed upside down, flipped, and rotated). The name of the object to grasp was revealed at this moment and the video recording started at the same time. The subject could then open his eyes, search for the object, and once he found it he grasped it. After a few seconds the recording was stopped. Figure 5 shows a subject performing the experiment.

#### 5.1.2. Videos

We recorded 151 videos with our experimental setup 5.1.1. Eight types of objects with simple shape and identifiable color were used, see examples in figure 5. The duration of videos was between 3,6 s up to 11,9 s, that is  $6,5 \pm 0,9$  s on average. These videos are short as they depict the initiation of hand motion and grasping of the object-of-interest. They are split between Train (60%), Validation (20%) and Test (20%) sets as shown in table 2.









This video dataset is rather simple in a sense that the object and the background are well separable; there are no occlusions. Some motion blur is observed when the subject moved his head when searching for the object-of-interest. The real challenge of this dataset comes from the semi-automatic ground truth annotation and distractors. Indeed some frames are miss-annotated: the object in the video can have an incorrect label as the subject was distracted and did not fixate the right object. The localization can be inaccurate; the bounding box can be too big or too small due to distance measurement inaccuracy in saliency map computation.

#### 5.2. Semi Automatic Annotation

Annotation is the process of describing the content of a set of videos, in each frame of each video, for all type of content (objects). In computer vision, this is done manually: a human annotator visualizes the videos, select objects in each frame depicting their bounding boxes (rectangle+label) and sometimes segments the object (binary mask+label). Datasets are now on the order of millions of images, and thousand of object types turning annotation into enormous and tedious work.

In our experiment we know the subject looked at the object, so we propose to use the saliency map to select the patch of the object (the saliency peak is located on the object). We threshold the saliency to create an approximate segmentation mask. In practice, there is a delay between the beginning of the video and the moment the subject opens his eyes, and finds the object of interest, called visual exploration (see

Table 2: Number of videos in the LEGO dataset, by category for Train, Validation and Test sets.

	Categories	Training	Validation	Testing	Total
	Background	90	31	0	121
	Cone	13	5	4	22
	Cylinder	4	2	1	7
	Hemisphere	8	3	3	14
	Hexagonal_Prism	10	4	3	17
	Rectangular_Prism	17	5	6	28
	Rectangular_Pyramid	10	4	3	17
	Triangular_Prism	17	5	4	26
	Triangular_Pyramid	11	3	4	18
	<b>Total/BGD</b>	<b>90</b>	<b>31</b>	<b>28</b>	<b>149</b>

section 3.1. It is on the order of 300 ms. We have to ignore this part of the sequence or we would be considering patches of objects other than the object of attention.

To solve this problem we developed a simplified annotation tool presented in figure 6, that allows the human annotator **(1)** to select the moment when the scene exploration is completed and the subject is focused on the object-of-interest (buttons 7 and 8), **(2)** to threshold the saliency map (button 9), and **(3)** to choose the category of the object (button 10). We allow the threshold to be changed because in some sequences the saliency map is larger due to imprecision in the distance to the object-of-interest computed by the eye-tracker software. Changing the thresholds allows the annotator to control the amount of context inside the object patch.

When the subject is distracted by other objects or lightning changes in the background the human annotator is not always able to detect this moment. This leads to the noise in training data and selection of a "bad" object proposal in online test scenario. This results in an accuracy drop we observed in our experiments. We deal with it using score fusion (see section 4.3)

### 5.3. Patch Extraction

We extracted an image patch as described in section 4.2.1. We only took one background patch on frames showing an object in the training and validation dataset, so that the final number of background patch sums up almost to the sum of those in object categories. We do not sample background on the test set. Ultimately, we had as many patches per category as the ImageNet dataset [2], but fewer categories were considered: 9 instead of 1000.

### 5.4. Network Optimization

We used the ImageNet architecture [25], changing the number of outputs of the last FC layer to match our number of classes. The learning rate was set to 0,001 and was decreased every 30000 iterations by half.

The training loss is rapidly decreasing as shown in figure 7:left The validation accuracy rapidly reaches  $\sim 80\%$  as shown in figure 7:right indicating a stable training of our network. This fast training is due to the simplicity of our objects and the lack of clutter in the scenes.

Figure 6: Annotation Tool user interface: the sequence can be played using buttons 2 to 4; the visualization method and resolution can be selected with buttons 5 and 6; the annotation parameters with buttons 7 to 11.



Table 3: Number of image patches by category for Train, Validation and Test extracted from the LEGO dataset.









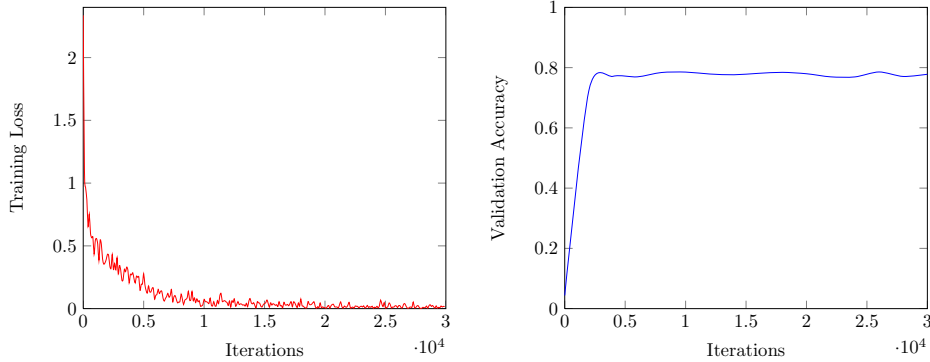
	Categories	Training	Validation	Testing	Total
	Background	123 424	43 024	0	166 448
	Cone	17 824	6 352	420	24 596
	Cylinder	6 544	2 928	111	9 583
	Hemisphere	13 360	4 016	272	17 648
	Hexagonal_Prism	16 592	5 776	235	22 603
	Rectangular_Prism	24 032	6 816	620	31 468
	Rectangular_Pyramid	10 736	4 448	308	15 492
	Triangular_Prism	21 168	7 744	396	29 308
	Triangular_Pyramid	16 784	4 976	412	22 172
	<b>Total</b>	<b>250 464</b>	<b>86 080</b>	<b>2 774</b>	<b>339 318</b>

Figure 7: Training and validation of the network. Left: Training loss and learning rate as a function of the number of iterations. Right: Validation accuracy as a function of the number of iterations



We train our network on a server equipped with 56 Intel Xeon cores and a NVIDIA Tesla k40m, it took two days to complete 47000 iterations.

### 5.5. Classification results

Figure 8 shows the average precision on all our categories as well as the gains of our temporal score fusion method. The deep CNN classifier was able to achieve the state-of-the-art performance with a mean average precision of 58,4%. Then score fusion presented in section 4.3 was able to recover the correct category over a video sequence even if less than 40% of the patches extracted on the frames were correctly classified. This was because the few correctly classified object proposals had strong score of their class. In the present work, we did fusion of scores on the whole video. For real time application in the assistive neuro-prosthesis visual system, the size of the fusion buffer would have to be optimized with respect to the latency of other components of the whole system. This leads to a score of 64,6% mAp, yielding a 6,2% gain.

We have conducted a performance test under Ubuntu 14.04 on an Intel I7-4790@3.6GHz CPU and a NVIDIA Quadro K4200 GPU. Computation of the saliency map from raw eye-tracking data was implemented in C++ with OpenCV and CUDA. This computation took 20 ms per frame without CUDA acceleration and 5 ms with it. We use the Caffe toolbox. It is not configured with CUDNN acceleration, so better performance can be expected with it. Classifying 2767 patches from 28 videos took 23.877 s which means 8.6 ms per patch, for a single video it means a computational time of 995 ms on average.

The total time for the computation of the saliency, and of classification of a patch, is of 28.6ms which is less than video frame rate (40ms) and much less than our requirement to be faster than a gaze-fixation time (250ms).

## 6. Discussion, conclusion and perspectives

In this work, we have proposed an approach for object recognition in egocentric videos guided by visual saliency to help grasping actions for neuro-prostheses. For annotation of visual data for training of object detectors in such a setting we also

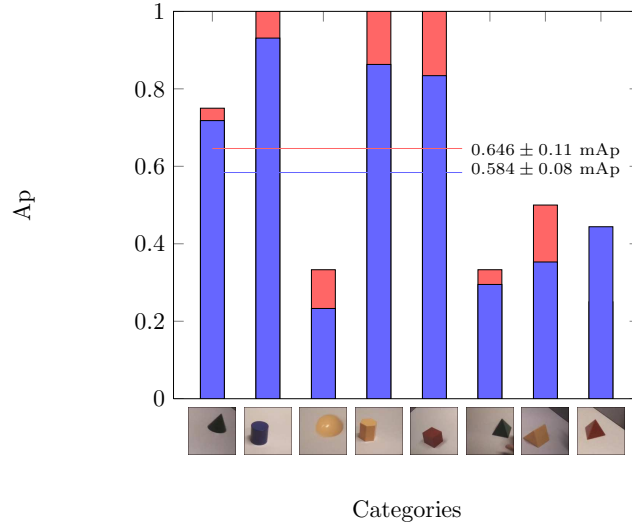


Figure 8: Average precision per class without (0,584 mAp), and with (0,646 mAp) score fusion

proposed a semi-automatic annotation method, guided by visual saliency as well. The recorded egocentric dataset will soon be made available online. Object recognition is performed using a deep CNN [25] that was able to achieve 58,4% mAp, and using temporal fusion of scores, we obtained state-of-the art results of 64,6% mAp despite the presence of annotation noise introduced by the distractors in training set and in the real-world "online" testing. The total time of our recognition system is about 28 ms per frame including visual saliency map computation and generalization with the Deep CNN. This time matches our requirement to be faster than visual fixation time.

This method has yet to be tuned for live system integration and some parameters such as the buffer size for temporal filtering have to be adjusted wrt the latency of other components of the neuro-prosthesis system. We are also eager to try other deeper and wider CNN architectures. The proposed approach gives rise to a wide set of exploration routes. Indeed, in such settings, the presence of noise in both annotation and test data sets are unavoidable due to human errors and physiology of human attention. We are thus interested in developing noise-robust optimization methods for Deep Neural Networks.

## Acknowledgments

This work was supported by CNRS-Idex grant PEPS Suivipp 2015.

## References

### References

- [1] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes



Challenge 2012 (VOC2012) Results, <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.

- [2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. M. S. Satheesh, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge (2015). doi:10.1007/s11263-015-0816-y.
- [3] C. Crispim-Junior, K. Avgerinakis, V. Buso, G. Meditskos, A. Briassouli, J. Benois-Pineau, Y. Kompatsiaris, F. Bremond, Semantic event fusion of different visual modality concepts for activity recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (99) (2016) 1–1. doi:10.1109/TPAMI.2016.2537323.
- [4] H. Pirsiavash, D. Ramanan, Detecting activities of daily living in first-person camera views, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, IEEE, 2012.
- [5] S. Chaabouni, J. Benois-Pineau, O. Hadar, C. B. Amar, Deep learning for saliency prediction in natural video. URL <https://hal.archives-ouvertes.fr/hal-01251614>
- [6] P. Parker, K. Englehart, B. Hudgins, Myoelectric signal processing for control of powered limb prostheses, *J Electromyogr Kinesiol* 16 (6) (2006) 541–548.
- [7] L. H. Smith, L. J. Hargrove, B. A. Lock, T. A. Kuiken, Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay, *IEEE Trans Neural Syst Rehabil Eng* 19 (2) (2011) 186–192.
- [8] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, O. C. Aszmann, The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges, *IEEE Trans Neural Syst Rehabil Eng* 22 (4) (2014) 797–809.
- [9] J. M. Hahne, F. Biessmann, N. Jiang, H. Rehbaum, D. Farina, F. C. Meinecke, K. R. Muller, L. C. Parra, Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control, *IEEE Trans Neural Syst Rehabil Eng* 22 (2) (2014) 269–279.
- [10] N. Jiang, K. B. Englehart, P. A. Parker, Extracting simultaneous and proportional neural control information for multiple-DOF prostheses from the surface electromyographic signal, *IEEE Trans Biomed Eng* 56 (4) (2009) 1070–1080.
- [11] B. Wodlinger, D. M. Durand, Localization and recovery of peripheral neural sources with beamforming algorithms, *IEEE Trans Neural Syst Rehabil Eng* 17 (5) (2009) 461–468.
- [12] T. A. Kuiken, G. Li, B. A. Lock, R. D. Lipschutz, L. A. Miller, K. A. Stubblefield, K. B. Englehart, Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms, *JAMA* 301 (6) (2009) 619–628.
- [13] M. Markovic, S. Dosen, C. Cipriani, D. Popovic, D. Farina, Stereovision and augmented reality for closed-loop control of grasping in hand prostheses, *J Neural Eng* 11 (4) (2014) 046001.

- [14] M. Markovic, S. Dosen, D. Popovic, B. Graimann, D. Farina, Sensor fusion and computer vision for context-aware control of a multi degree-of-freedom prosthesis, *J Neural Eng* 12 (6) (2015) 066022.
- [15] E. A. Corbett, E. J. Perreault, K. P. Kording, Decoding with limited neural data: a mixture of time-warped trajectory models for directional reaches, *J Neural Eng* 9 (3) (2012) 036002.
- [16] E. A. Corbett, N. A. Sachs, K. P. Kording, E. J. Perreault, Multimodal decoding and congruent sensory information enhance reaching performance in subjects with cervical spinal cord injury, *Front Neurosci* 8 (2014) 123.
- [17] R. R. Kaliki, R. Davoodi, G. E. Loeb, Evaluation of a noninvasive command scheme for upper-limb prostheses in a virtual reality reach and grasp task, *IEEE Trans Biomed Eng* 60 (3) (2013) 792–802.
- [18] R. B. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (1) (2016) 142–158. doi:10.1109/TPAMI.2015.2437384. URL <http://dx.doi.org/10.1109/TPAMI.2015.2437384>
- [19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, *CoRR abs/1312.6229*. URL <http://arxiv.org/abs/1312.6229>
- [20] P. Agrawal, R. B. Girshick, J. Malik, Analyzing the performance of multilayer neural networks for object recognition in: *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII, 2014*, pp. 329–344. doi:10.1007/978-3-319-10584-0\_22. URL [http://dx.doi.org/10.1007/978-3-319-10584-0\\_22](http://dx.doi.org/10.1007/978-3-319-10584-0_22)
- [21] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013, 2013*, pp. 3626–3633. doi:10.1109/CVPR.2013.465. URL <http://dx.doi.org/10.1109/CVPR.2013.465>
- [22] J. Uijlings, K. V. de Sande, T. Gevers, A. Smeulders, Selective search for object recognition, *International Journal of Computer Vision* 104 (2) (2013) 154–171. URL <https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013>
- [23] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Is object localization for free? - weakly-supervised learning with convolutional neural networks, in: *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR), 2015, Boston, MA, USA, June 7-12, 2015, 2015*, pp. 685–694. doi:10.1109/CVPR.2015.7298668. URL <http://dx.doi.org/10.1109/CVPR.2015.7298668>

- [24] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (9) (2015) 1904–1916.  
doi:10.1109/TPAMI.2015.2389824.  
URL <http://dx.doi.org/10.1109/TPAMI.2015.2389824>
- [25] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 2012, pp. 1106–1114.  
URL <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>
- [26] R. B. Girshick, Fast R-CNN, CoRR abs/1504.08083.  
URL <http://arxiv.org/abs/1504.08083>
- [27] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching., in: *CVPR*, IEEE Computer Society, 2007.  
URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2007.html#PhilbinCISZ07>
- [28] R. de Carvalho Soares, I. da Silva, D. Guliato, Spatial locality weighting of features using saliency map with a bovw approach, in: *International Conference on Tools with Artificial Intelligence*, 2012, 2012, pp. 1070–1075.  
doi:10.1109/ICTAI.2012.151.
- [29] I. González-Díaz, V. Buso, J. Benois-Pineau, Perceptual modeling in the problem of active object recognition in visual scenes, *Pattern Recognition* 56 (2016) 129–141. doi:10.1016/j.patcog.2016.03.007.  
URL <http://dx.doi.org/10.1016/j.patcog.2016.03.007>
- [30] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (9) (2010) 1627–1645.  
doi:10.1109/TPAMI.2009.167.  
URL <http://dx.doi.org/10.1109/TPAMI.2009.167>
- [31] C. Chamaret, J. Chevet, O. L. Meur, Spatio-temporal combination of saliency maps and eye-tracking assessment of different strategies.
- [32] S. Martinez-Conde, S. L. Macknik, X. G. Troncoso, D. H. Hubel, Microsaccades: a neurophysiological analysis, *Trends in neurosciences* 32 (9) (2009) 463–475.
- [33] H. Devillez, A. Gurin-Dugu, N. Guyader, How task difficulty influences eye movements when exploring natural scenes, in: *EUSIPCO*, IEEE, 2015, pp. 1536–1540.  
URL <http://dblp.uni-trier.de/db/conf/eusipco/eusipco2015.html#DevillezGG15>
- [34] D. Villani, F. Morganti, P. Cipresso, S. Ruggi, G. Riva, G. Gilli, Visual exploration patterns of human figures in action: an eye tracker study with art paintings, *Front Psychol* 6 (2015) 1636.
- [35] D. Wooding, Fixation maps: Quantifying eye-movement traces, in: *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02, ACM, New York, NY, USA, 2002, pp. 31–36. doi:10.1145/507072.507078.  
URL <http://doi.acm.org/10.1145/507072.507078>

- [36] J. Yangqing, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22Nd ACM International Conference on Multimedia, MM '14, ACM, New York, NY, USA, 2014, pp. 675–678. doi:10.1145/2647868.2654889. URL <http://doi.acm.org/10.1145/2647868.2654889>